


## STILL PICTURE EXTRACTION DEVICE AND PROGRAM STORAGE MEDIUM FOR EXTRACTING STILL PICTURE

Patent Number: JP2000261741  
Publication date: 2000-09-22  
Inventor(s): NAKAGAWA TOSHIO; ITO TAKAYUKI; SEKI  
TAKAO  
Applicant(s): NIPPON HOSO KYOKAI  
Requested Patent:  JP2000261741  
Application  
Number: JP19990057592 19990304  
Priority Number(s):  
IPC Classification: H04N5/76; H04N5/262; H04N5/91  
EC Classification:  
Equivalents:

---

### Abstract

---

**PROBLEM TO BE SOLVED:** To automatically extract a still picture as a summary picture without manual intervention from a moving picture including an operation of a human body and an object.

**SOLUTION:** In a video image editor-image processing unit or the like employing a nonlinear editor or a computer, a motion of a pixel area or a characteristic point in a moving picture including a motion of an object or a human body is measured, and a point of time when the motion is temporally discontinuous is detected (2, 4, 6, 8, 10), a still picture at that point of time is outputted from a frame number and summarized picture extract section 12 so as to automatically extract a summarized picture representing contents of the operation included in the moving picture.

---

Data supplied from the esp@cenet database - 12

THIS PAGE BLANK (USPTO)



## 【特許請求の範囲】

【請求項 1】 動画像情報を入力する入力手段と、入力された前記動画像における動きが時間的に不連続となる時点を検出する不連続点検出手段と、前記不連続となる時点の動画像に基づいた静止画像を形成する出力手段とを具備したことを特徴とする静止画抽出装置。

【請求項 2】 請求項 1 に記載の静止画抽出装置において、前記出力手段は、動画像の静止要約画像として、一枚の静止画像、または複数の静止画像を縮小して統合した一枚の静止画像を形成することを特徴とする静止画抽出装置。

【請求項 3】 請求項 1 に記載の静止画抽出装置において、さらに加えて、不連続を検出した動画像の時刻を表すタイムコードを出力する手段を備えたことを特徴とする静止画抽出装置。

【請求項 4】 請求項 1 ないし請求項 3 のいずれかに記載の静止画抽出装置において、前記出力手段は、形成された静止画を可視化するための表示手段または印刷手段を含むことを特徴とする静止画抽出装置。

【請求項 5】 動画像から静止画像を抽出するためのプログラムを記憶した記憶媒体であって、動画像情報を入力するステップと、入力された前記動画像における動きが時間的に不連続となる時点を検出するステップと、前記不連続となる時点の動画像に基づいた静止画像を形成するステップとを含んだ制御手順を、読み出し可能なプログラムの形態で記憶したことを特徴とする記憶媒体。

【請求項 6】 請求項 5 に記載の記憶媒体において、動画像の静止要約画像として、一枚の静止画像、または複数の静止画像を縮小して統合した一枚の静止画像を形成するステップを含んだことを特徴とする記憶媒体。

【請求項 7】 請求項 5 に記載の記憶媒体において、不連続を検出した動画像の時刻を表すタイムコードを出力するステップを含んだことを特徴とする記憶媒体。

【請求項 8】 請求項 5 ないし請求項 7 のいずれかに記載の記憶媒体において、形成された静止画を可視化するための表示ステップまたは印刷ステップを含んだことを特徴とする記憶媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、動画像中から特定の静止画像を抽出する静止画抽出装置、および、静止画抽出のためのプログラム記憶媒体に関するものである。

【0002】 さらに詳述すると、本発明は、例えばノンリニア編集機や計算機を使った映像編集装置・動画像データ処理装置において、多くの動画像を人間が閲覧・選択する際に使われる一覧用の静止画による要約画像を生成するのに好適な、静止画抽出装置および静止画抽出のためのプログラム記憶媒体に関するものである。また、本発明は、動画像の概要を紙媒体上などに静止画データとして表現するのに好適な、静止画抽出装置および静止

画抽出のためのプログラム記憶媒体に関するものである。

## 【0003】

【従来の技術】 従来から、動画像の一覧用の要約画像を作成するには、

①動画像カット先頭のフレームや同一のカットの時間的中心点などを内容に関わらず選択するか：

②人手によりカットの内容を確認し、動画像の内容を良く表現していると思われる時点の画像を明示的に指定する：方法が採られていた。

【0004】 しかし、動画像の内容を代表する静止画を自動的に抽出するための装置は存在していなかった。

## 【0005】

【発明が解決しようとする課題】 従来技術として上記②に示した通り、人物などの動作が含まれた動画像カットから、動作の内容を端的に表すことのできる一または複数の静止画を自動的に抽出するには、この作業を人手によるしかなかったため、ノンリニア編集機や計算機上などでの動画像素材の閲覧用の要約画像を作成するには、多大な手間が必要とされた。

【0006】 よって本発明の目的は、上述の点に鑑み、人物や物体の動作が含まれた動画について、人手を介することなく、要約画像としての静止画を自動的に抽出するようにした静止画抽出装置および静止画抽出のためのプログラム記憶媒体を提供することにある。

## 【0007】

【課題を解決するための手段】 上記の目的を達成するために、本発明に係る静止画抽出装置は、動画像情報を入力する入力手段と、入力された前記動画像における動きが時間的に不連続となる時点を検出する不連続点検出手段と、前記不連続となる時点の動画像に基づいた静止画像を形成する出力手段とを具備したものである。

【0008】 ここで、上記の静止画抽出装置において、前記出力手段は、動画像の静止要約画像として、一枚の静止画像、または複数の静止画像を縮小して統合した一枚の静止画像を形成することが可能である。また、不連続を検出した動画像の時刻を表すタイムコードを出力する手段を備えることも可能である。さらに、前記出力手段は、形成された静止画を可視化するための表示手段または印刷手段を含むことも可能である。

【0009】 本発明に係る記憶媒体は、動画像から静止画像を抽出するためのプログラムを記憶した記憶媒体であって、動画像情報を入力するステップと、入力された前記動画像における動きが時間的に不連続となる時点を検出するステップと、前記不連続となる時点の動画像に基づいた静止画像を形成するステップとを含んだ制御手順を、読み出し可能なプログラムの形態で記憶したものである。

【0010】 ここで上記の記憶媒体において、動画像の静止要約画像として、一枚の静止画像、または複数の静

止画像を縮小して統合した一枚の静止画像を形成するステップを含むことができる。また、不連続を検出した動画像の時刻を表すタイムコードを出力するステップを含むことも可能である。さらに、形成された静止画を可視化するための表示ステップまたは印刷ステップを含むことも可能である。

【0011】上記の構成を有する静止画抽出装置および静止画抽出のためのプログラム記憶媒体によれば、例えば、ノンリニア編集機や計算機を使った映像編集装置・画像処理装置において、人物や物体の動作が含まれた動

【0012】

【発明の実施の形態】本発明では、人物や物体の動作が含まれる動画像カットの中で、ある一定時間または一定フレーム数以上の連続した一連の動きを行う画素領域またはエッジや交点などの特徴点を検出し、その動きが開始、または、終了する時点のフレームを要約画像として決定する。

【0013】連続した動きを行う画素領域や特徴点を求める手法については、一般的な画像処理分野における動き追跡手法を用いることができる。そして、それぞれの画素領域や特徴点の動きベクトルがある指定する値以上の角度および大きさの時間変化を行った場合、その時点で連続した動きは途切れるものとして判定し、その時点のフレームを要約画像とする。

【0014】要約画像と決定されたフレームは、静止画としてディスプレイなどに表示したり、画像データとして出力するほか、要約画像そのものを出力する代わりに対応するフレーム番号、タイムコードの形で出力しても良い。また、用途に応じて、フレームの替わりにフィールドなどの異なる動画像単位を用いることもできる。

【0015】以下、図面を参照して、本発明の実施の形態を詳細に説明していく。

【0016】図1は、本発明を適用した静止画抽出装置の全体構成を示す。本図において、2は映像データを入力するフレームバッファ部、4は動きベクトル検出部、6は動きベクトル領域化部、8は領域動き追跡部、10は領域動き始点・終点検出部、12はフレーム番号および要約画像抽出部である。また、14は本装置全体の動作を制御するための制御部、16は制御部14の制御手続き（後に、図4～図10において詳述する）を記憶してあるメモリ（ROMまたはRAM）である。

【0017】フレームバッファ部2においては、動画像データをフレームバッファに取り込む。

【0018】動きベクトル検出部4においては、各フレームの画像データと、その1フレーム前の画像データから、画像に含まれる動きベクトルを計算する。

【0019】次の動きベクトル領域化部6においては、各フレームにおいて動きベクトル検出部4で出力された動きベクトルのうち、画像中で座標的に隣接し、かつ、類似する動きベクトルをもつ画素を一つの領域にまとめる領域化を行う。

【0020】領域動き追跡部8においては、動きベクトル領域化部6で各フレームについてまとめられた類似する動きベクトルをもつ領域について、以前のフレームのどの領域と時間的に対応するかを検査する領域動き追跡処理を行い、ある動き領域が過去何フレームにわたって存在しているかを計算する。

【0021】領域動き始点・終点検出部10では、領域動き追跡部8で求められた動き領域の時間的遷移が途切れている場合、その持続時間のフレーム数が一定の時間の長さを越えるかどうか検査し、越える場合その時間的遷移の始点および終点を検出し、動画像から要約画像を抽出する時刻と決定する。

【0022】フレーム番号および要約画像抽出部12では、領域動き始点・終点検出部10で決定された時刻のデータを出力するか、また、元の動画像からその時刻の静止画を抽出した静止画を要約画像として出力する。

【0023】これら各部2、4、6、8、10、12における処理を動画像が1フレーム入力される毎に繰り返すことにより、動画像から要約画像もしくは要約画像を示すタイムコードや時刻データを出力する。

【0024】次に、これら各部2、4、6、8、10、12の動作を詳細に説明する。

【0025】フレームバッファ部2について

フレームバッファ部2では、図2に示すように、時間と共に入力される動画像を前後2フレーム分、各フレームバッファ22、23に蓄積する。動画像は、ビデオ入力をA/D変換部20でA/D変換して入力するか、または、直接デジタル動画像データの形で入力し、フレーム毎にフレームバッファ22、23に蓄積する。

【0026】すなわち、図2に示すように、時刻 $t$ のフレームが入力された時点では、そのフレームの画像データはフレームバッファ $F(0)$ に、時刻 $t-1$ のフレームの画像データはフレームバッファ $F(1)$ に蓄積される。

【0027】動きベクトル検出部4について

動きベクトル検出部4では、図3に示すように、フレームバッファ $F(1)$ の画像からフレームバッファ $F(0)$ への画像間の各画素毎の動きベクトルを動き検出部40で求め、その結果を動きベクトル用バッファ41に格納する。この動きベクトルの計算は、一般に使用されるブロックマッチング法や勾配法などの手法を、ソフトウェア的に構成するか、市販されている画像の動き検出LSIなどを使用してハードウェア的に構成して実現する。ここでは、ブロックマッチング法による例を示す。

【0028】いま、フレームバッファF(0)の画像における座標(x, y)の画素の値をI0(x, y)、フレームバッファF(1)の画像における座標(x, y)の画素の値をI1(x, y)とする。F(0)の画像の中で、ある定められた辺の長さZをもち、左上隅の座標が(x, y)の正方の画像小部分を定義し、その小部分の明るさのパターンが、F(1)の画像の中のどの位置の同じ大きさの画像小部分と最も類似しているかを計算することで動きベクトルが求められる。

$$\sum_{dy=0}^{Z-1} \sum_{dx=0}^{Z-1} \{(I0(x+dx, y+dy) - I1(x+dx-vx, y+dy-vy))^2\}$$

【0031】ただし、kはあらかじめ定められた定数とする。

【0032】このようにして、動きベクトル検出部4において、座標(x, y)における2つのフレーム間の動きベクトルV(x, y)が各画素毎に求められる。

【0033】動きベクトル領域化部6について  
動きベクトル領域化部6では、動きベクトル検出部4で求められた各画素毎の動きベクトル(x, y)を類似したものの同士まとめ、それらのベクトルをもつ画素に同じラベルをつけていく領域化処理を行う。ここで、座標(x, y)をベクトル表記し、pとおき、各画素の動きベクトルを、v(p)と表記上置き換えておく。

【0034】

【数2】 $v(p) = V(x, y)$

また、以降、画素pと表記したもののについては、以降、座標としてpをもつ画素を示すものとする。さらに、各画素ベクトルがどの領域に属するかを区別するためのラベルを格納するためにラベルバッファを用意する。ここでは、ラベルバッファの中で画素pのラベルは、l(p)にラベルの番号として格納されるとする。

【0035】動きベクトル領域化部6の具体的な処理は、図4～図6の流れ図に示される次のような処理を行う。

【0036】ステップS1：画像中のすべての画素pについて、各画素がどの領域にも属していない状態、すなわち、ラベルをついていない状態にする。ここでは、ラベルの値が0の場合はラベルがつけられていないことを意味すると定義し、 $l(p) = 0$ として初期化しておく。また、この処理で今までにつけられたラベルの最大番号を示す変数Lmaxを初期値0として用意する。以降の処理で、各画素に新しいラベルがつけられる場合には、Lmaxを1ずつ増やし、そのLmaxの値を割り当てるものとする。すなわち、新たな領域のラベルは1から順に番号を増やしながつけられる。

【0037】ステップS2：画像の左上隅より順に、以下のステップS3～ステップS5A, 5B, 5C, 5D, 5Eの処理を行う。

【0038】ステップS3：注目する画素をpとし、画

\*【0029】例えば、F(0)のある画像小部分J(x, y)が、座標(x, y)と座標(x+z-1, y+z-1)をそれぞれ左上隅と右下隅として持つ場合、このJ(x, y)に対する動きベクトルV(x, y)(x成分にvx, y成分にvyをもつベクトルとする)は、以下の式を最小にするvx, vyを-k～+kの範囲内でそれぞれ見つけることで求められる。

【0030】

【数1】

素pの動きベクトルがある一定の大きさαより大きな場合、すなわち、 $|v(p)| > \alpha$ の時、以下のステップS4～ステップS5A, 5B, 5C, 5D, 5Eの処理を行う。

【0039】ステップS4：画素pの隣接する画素qに対して、動きベクトルv(p)とv(q)の間での類似性の有無を、例えば以下のような条件式が満たされるかどうかで判断する。

【0040】

【数3】 $\|v(p) - v(q)\| < \beta$   
 $(v(p) \cdot v(q)) / (\|v(p)\| \|v(q)\|) > \gamma$

ここで、β, γはパラメータとなる定数を示す。

【0041】この条件式が双方満たされる場合は、ベクトルv(p)とv(q)に類似性があるとみなす。

【0042】ステップS5A, 5B, 5C, 5D, 5E：ステップS4の条件が画素pと画素pに隣接する上下左右の4つのすべての画素(それぞれq1, q2, q3, q4とする)について満たされる場合、画素pおよび4近傍の画素のラベルを同じ値に設定する。手順としては、例えば次のような処理を行う。

【0043】もし、 $l(q1) \neq 0$ (ラベルがついている)なら、

【0044】

【数4】 $l(p) = l(q1)$

$l(q2) = l(q1)$

$l(q3) = l(q1)$

$l(q4) = l(q1)$

の代入を行い、それ以外の場合、Lmaxを1増やして新しいラベルとして、

【0045】

【数5】 $l(p) = Lmax$

$l(q1) = Lmax$

$l(q2) = Lmax$

$l(q3) = Lmax$

$l(q4) = Lmax$

というように代入を行う。

【0046】ステップS3からステップS5A, 5B,

5C, 5D, 5Eの処理をすべての画素について行うことによって、画像中にいくつかのラベルのついた領域(同じラベルのついた画素の集合)が $L_{max}$ 個、結果として得られる。

【0047】ステップS6:ここまでで得られた $L_{max}$ 個の領域を $T(1) \sim T(L_{max})$ と表し、ここで各領域の面積を各ラベルのついた画素の個数を数えることで計算する。具体的には、各領域の面積を $S(1) \sim S(L_{max})$ とすると、 $S(1) \sim S(L_{max})$ を初期化で0にしておいた後に、すべての画素 $p$ について、 $S(1(p)) = S(1(p)) + 1$ を行うことで、面積が計算できる。

【0048】ステップS7:領域 $T(1) \sim T(L_{max})$ について、それぞれの領域に含まれる動きベクトルの平均を求め、それぞれ $Av(1) \sim Av(L_{max})$ とする。

【0049】以上により、動きベクトル領域化部6において、ある時刻のフレームにおける動き領域が $L_{max}$ 個求められる。

【0050】領域の個数 $L_{max}$ 、1から $L_{max}$ 番のラベルをつけられた領域の面積 $S(1) \sim S(L_{max})$ と動き平均ベクトル $Av(1) \sim Av(L_{max})$ 、そして、どの画素がどの番号の領域に属するかを表す変数 $l(p)$ が1つのフレームに対する動きベクトル領域化部6の出力となる。

#### 【0051】領域動き追跡部8について

図7～図9に示すステップS11～ステップS21は、領域動き追跡部8における処理を示す。

【0052】領域動き追跡部8では、動きベクトル領域化部6で出力される情報を1フレーム分蓄積しておき、ある時刻に動きベクトル領域化部6から出力された情報と1フレーム前の情報から、現在の個々の動き領域が、1フレーム前のどの動き領域と時間的に対応するかを検査する。

【0053】ある時刻 $t$ において、動きベクトル領域化部6から出力された動き領域 $T(1) \sim T(L_{max})$ の情報が出力される。情報としては、動きベクトル領域化部6で使われた見つけられた領域の個数 $L_{max}$ 、それぞれの領域の面積 $S(1) \sim S(L_{max})$ 、それぞれの領域の動き平均ベクトル $Av(1) \sim Av(L_{max})$ 、および、各画素 $p$ が何番の領域に属するかを示した変数 $l(p)$ が領域動き追跡部8に入力される。

【0054】さらに領域動き追跡部8では、動き領域が過去何フレームにわたって画像中に存在していたかを表す変数 $N(1) \sim N(L_{max})$ を用意しておく。

【0055】また、1フレーム前の情報を比較するために、上記、 $L_{max}$ 、 $T(1) \sim T(L_{max})$ 、 $S(1) \sim S(L_{max})$ 、 $Av(1) \sim Av(L_{max})$ 、 $l(p)$ 、 $N(1) \sim N(L_{max})$ に対応する1フレーム前の情報をそれぞれ、 $L_{max}'$ 、 $T'$

$(1) \sim T'(L_{max}')$ 、 $S'(1) \sim S'(L_{max}')$ 、 $Av'(1) \sim Av'(L_{max}')$ 、 $l'(p)$ 、 $N'(1) \sim N'(L_{max}')$ として蓄積してあるものとする。また、1フレーム前の領域 $T'(k')$ で、現在のフレームに対応する領域が見つかった場合のフラグを格納する変数として $F'(k')$ を用意しておく。 $F'(k')$ は初期値として0を入れておく。

【0056】ある領域 $T(k)$ が、1フレーム前のどの領域と対応するかを調べる次のような処理を $k=1 \sim L_{max}$ について行う。領域 $T(k)$ の面積 $S(k)$ がある一定値 $S0$ 以下の場合、以下の処理は行わず、 $N(k)=0$ として、次の領域に処理を移す。

【0057】領域の面積が $S0$ より大きい場合、領域 $T(k)$ を $-Av(k)$ だけ平行移動させた領域 $T0(k)$ を計算する。1フレーム前の領域 $T'(1) \sim T'(L_{max}')$ のうち $T0(k)$ と最も重なる面積が最も大きいものを検索し、 $T'(k')$ とする。

【0058】さらにその重なる面積と $T0(k)$ の面積( $S(k)$ に等しい)の比が、ある一定の割合 $\delta$ を越える場合、時刻 $t-1$ のフレームの動き領域 $T'(k')$ は、時刻 $t$ のフレームの動き領域 $T(k)$ と対応するものと判断し、領域 $T(k)$ の過去からの存在時間 $N(k)=N'(k')+1$ とする。

【0059】また、1フレーム前の領域は時刻 $t$ においても対応する領域が見つかったことを示すために、 $F'(k')=1$ とフラグを立てておく。上記面積比が $\delta$ 以下の場合、領域 $T(k)$ は時刻 $t$ において始めて現れた領域として、過去からの存在時間 $N(k)=1$ とする。

【0060】領域動き始点・終点検出部10について  
図10に示すステップS31～ステップS36は、領域動き始点・終点検出部10における処理を示す。

【0061】時刻 $t$ におけるすべての動き領域について上記の処理を行った後、領域動き始点・終点検出部10において、1フレーム前(時刻 $t-1$ )の領域 $T'(1) \sim T'(L_{max}')$ の中で、時刻 $t$ の領域と対応関係が見つけられない領域を $F'(1) \sim F'(L_{max}')$ の値が0で、かつ、過去の領域の存在時間 $N'(k')$ がある一定値 $A$ を越えるものを探し、見つけた場合、時刻 $t-N'(k')$ と、時刻 $t-1$ を動き領域 $T'(k')$ が開始および終了する時刻(領域の動きの不連続が発生する時刻)として、フレーム番号および要約画像抽出部12へ出力する。

【0062】フレーム番号および要約画像抽出部12について  
フレーム番号および要約画像抽出部12では、領域動き始点・終点検出部10より出力された時刻情報を、要約画像の存在する時刻データとしてそのまま出力するか、元の動画像からその時刻の静止画を要約画像として抽出し出力する。

## 【0063】

【発明の効果】以上説明した通り、本発明によれば、人物や物体の動作が含まれた動画について、人手を介すことなく、要約画像としての静止画を自動的に抽出することができる。

【0064】すなわち、人物や物体の動作が含まれた動画画像に対し、本発明によれば、従来は人手に頼らざるを得なかった動画画像の内容を代表する静止画像を要約画像として抽出する作業が自動的に行え、例えばノンリニア編集機や計算機を使った映像編集用の閲覧用要約画像を人の手間をかけず作成することができる。また、要約画像は紙媒体などに対しても印刷可能であるので、映像とは異なる紙などの媒体上で動作を表現できる。これらの効果により、番組や印刷物の作成における省力化と表現機能の向上を図ることができる。

## 【図面の簡単な説明】

【図1】本発明を適用した静止画抽出装置の全体構成を示すブロック図である。

【図2】フレームバッファ部2の詳細を示すブロック図である。

【図3】動きベクトル検出部4の詳細を示すブロック図である。

【図4】動きベクトル領域化部6における処理手順を示

すフローチャートである。

【図5】動きベクトル領域化部6における処理手順を示すフローチャートである。

【図6】動きベクトル領域化部6における処理手順を示すフローチャートである。

【図7】領域動き追跡部8における処理手順を示すフローチャートである。

【図8】領域動き追跡部8における処理手順を示すフローチャートである。

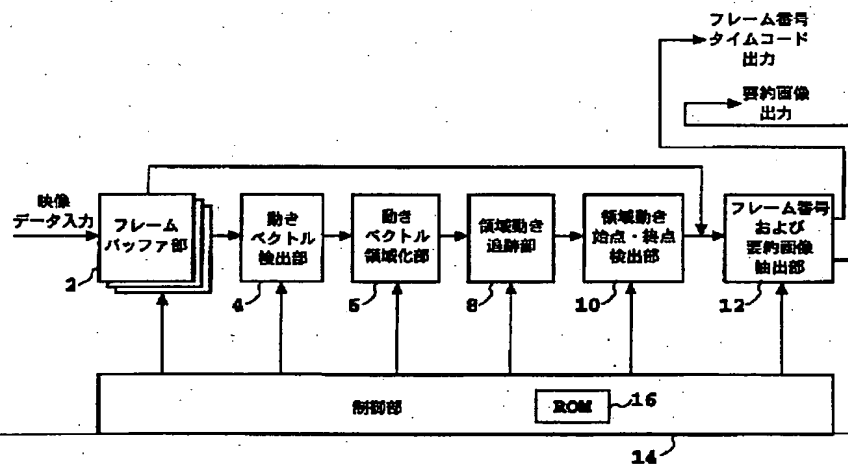
【図9】領域動き追跡部8における処理手順を示すフローチャートである。

【図10】領域動き始点・終点検出部10における処理手順を示すフローチャートである。

## 【符号の説明】

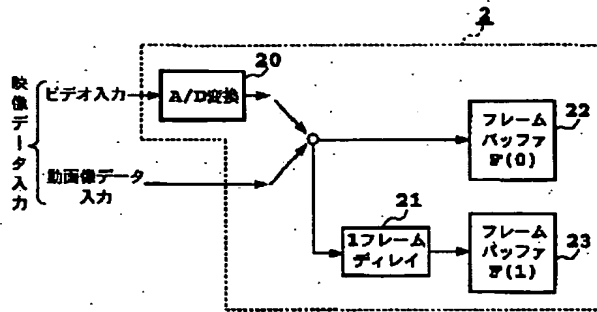
- 2 フレームバッファ部
- 4 動きベクトル検出部
- 6 動きベクトル領域化部
- 8 領域動き追跡部
- 10 領域動き始点・終点検出部
- 12 フレーム番号および要約画像抽出部
- 14 制御部
- 16 制御部14の制御手続きを記憶してあるメモリ（ROMまたはRAM）

【図1】

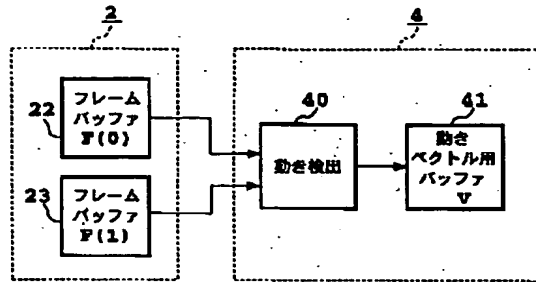




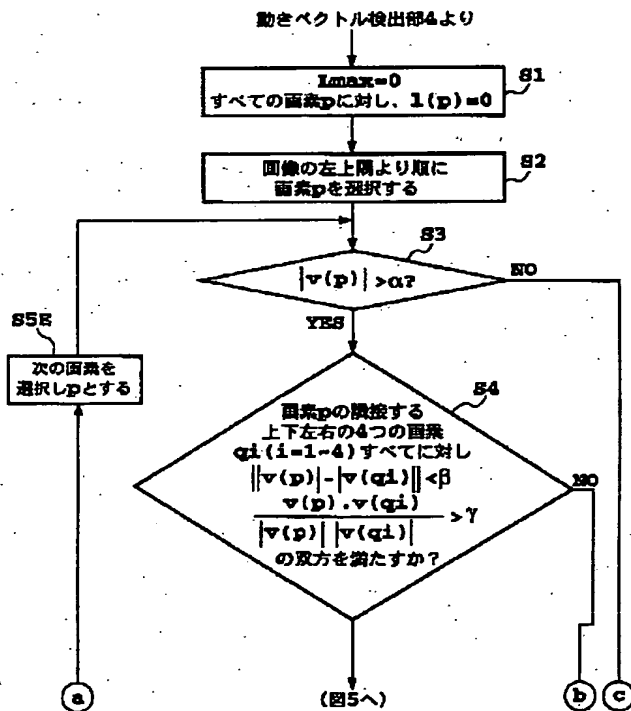
【図2】



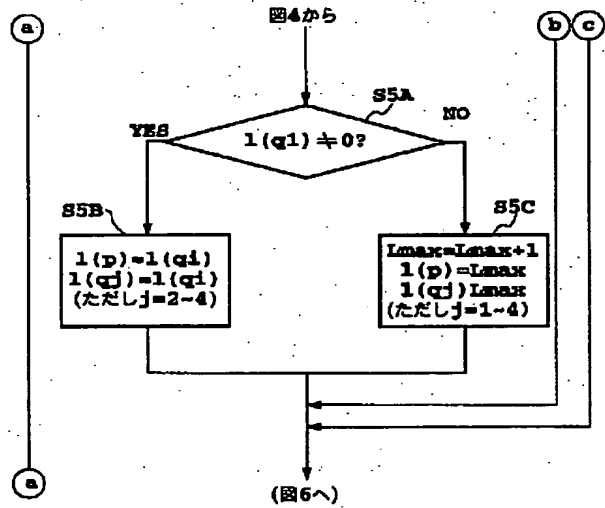
【図3】



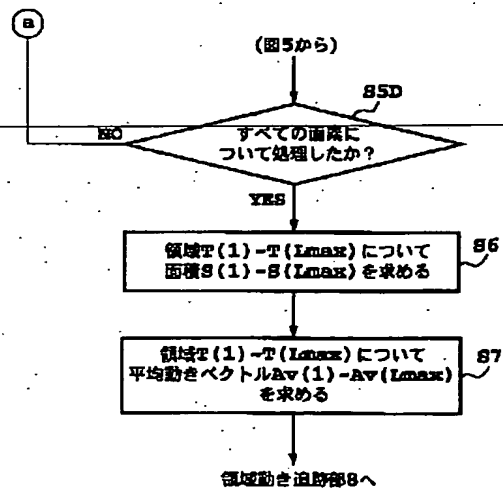
【図4】



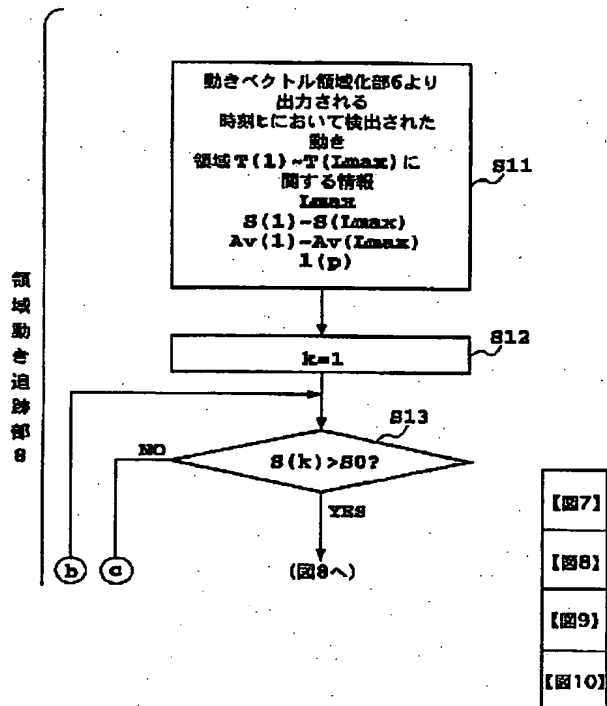
【図5】



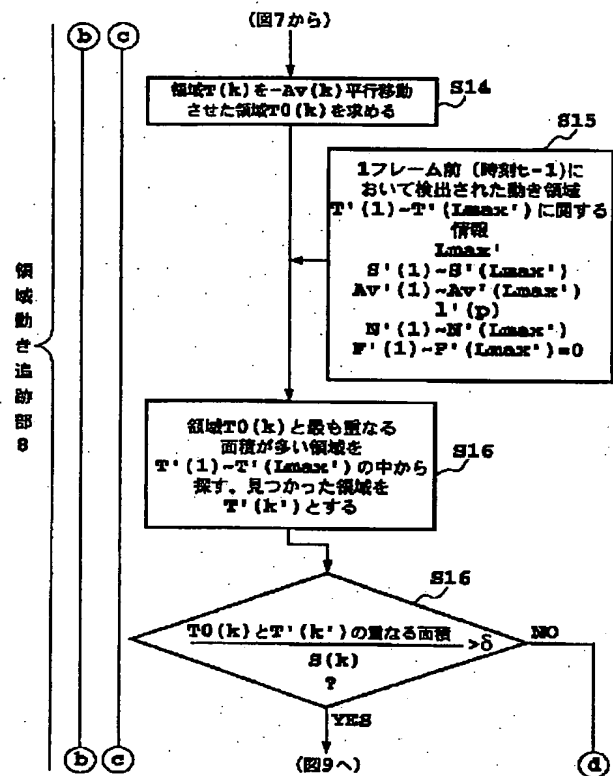
【図6】



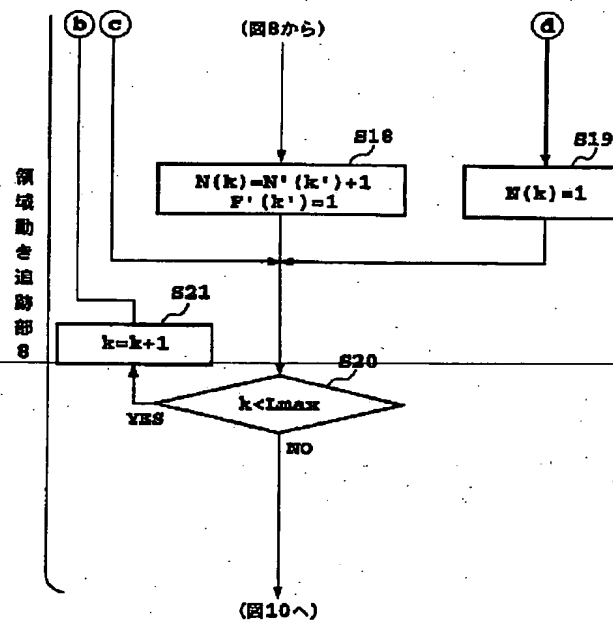
【図7】



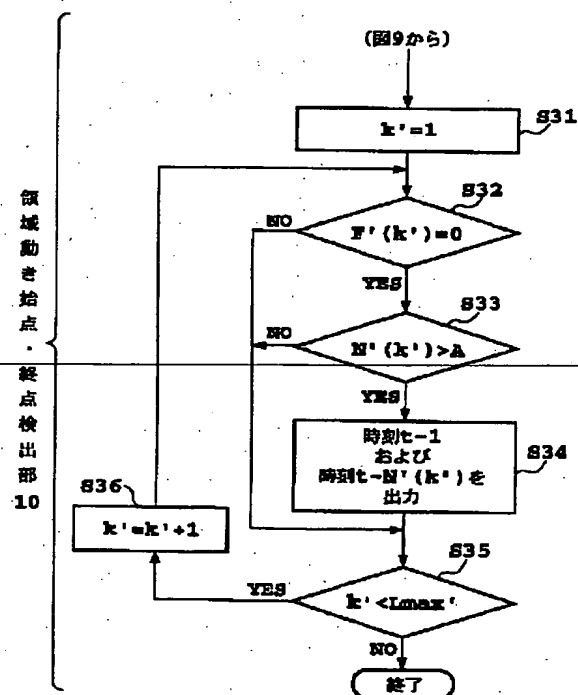
【図8】



【図9】



【図10】



フロントページの続き

Fターム(参考) 5C023 AA02 AA14 AA31 AA34 AA38  
BA04 BA11 CA02 DA02 DA08  
EA03 EA05 EA06  
5C052 AA11 AA17 AB04 AC08 DD04  
DD10  
5C053 FA04 FA14 FA21 FA27 HA29  
JA22 KA04 KA22 KA24 KA25  
LA03 LA06

---

**THIS PAGE BLANK (USPTO)**